AMENDMENTS TO THE CLAIMS

Kindly amend claims 1, 11, and 21 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A microprocessor apparatus, for precluding a pipeline stall due to microcode ROM access delay, the microprocessor apparatus comprising:

a translator, configured to generate a plurality of micro instruction queue entries, each of said plurality of micro instruction queue entries corresponding to an instruction, and said each of said plurality of micro instruction queue entries comprising a plurality of micro instructions and a microcode entry point, wherein said translator generates said each of said plurality of micro instruction queue entries in order;

a micro instruction queue, coupled to said translator, configured to receive said each of said plurality of micro instruction queue entries in said order, and configured to provide said each of said plurality of micro instruction queue entries to register logic in said ~~order, and order~~;

a microcode ROM, configured to provide a second part of a micro instruction sequence corresponding to said microcode entry point; and

early access logic, coupled to said micro instruction queue, configured to employ said microcode entry point to access a ~~microcode~~ said microcode ROM prior to when said microcode entry point is provided to said register logic, whereby said microcode ROM provides a first micro instruction from said second part to said register logic when said first micro instruction is required by said register logic, and wherein said early access logic employs said microcode entry point when said microcode entry point is within a bottom micro instruction queue entry, said bottom micro instruction queue entry comprising one of said each of said plurality of micro instruction queue entries, and wherein said bottom micro instruction queue entry will be provided to said register logic during a next clock cycle.

2. (Cancelled)

3. (Previously Presented) The microprocessor apparatus as recited in claim 1, wherein said plurality of micro instruction queue entries comprises four micro instruction queue entries.

4. (Original) The microprocessor apparatus as recited in claim 1, wherein said plurality of micro instructions comprises three micro instructions.

5. (Original) The microprocessor apparatus as recited in claim 4, wherein the microcode ROM access delay comprises four clock cycles.

6. (Cancelled)

7. (Cancelled)

8. (Previously Presented) The microprocessor apparatus as recited in claim 1, wherein said translator is configured to provide a generated micro instruction queue entry to a top micro instruction queue entry, wherein said top micro instruction queue entry comprises one of said each of said plurality of micro instruction queue entries.

9. (Previously Presented) The microprocessor apparatus as recited in claim 1, wherein said translator is configured to provide a generated micro instruction queue entry to a mux, and, when said plurality of micro instruction queue entries is empty, said mux provides said generated micro instruction queue entry to said register logic during a next clock cycle.

10. (Original) The microprocessor apparatus as recited in claim 9, wherein said early access logic employs a bypass microcode entry point corresponding to said generated micro instruction queue entry.

11. (Currently Amended) An apparatus for absorbing pipeline stalls associated with microcode ROM access delay, the apparatus comprising:

a micro instruction queue, for receiving a plurality of queue entries in order from a translator, and for providing said plurality of queue entries to register logic in said order, each of said plurality of queue entries comprising:

first micro instructions, all of said first micro instructions corresponding to an instruction; and

a microcode entry point, coupled to said first micro instructions, ~~configured to point to second micro instructions stored within a microcode ROM~~; and

a microcode ROM, configured to provide second micro instructions stored therein, and pointed to by said microcode entry points, and

early access logic, coupled to said micro instruction queue, configured to employ said microcode entry point to access said microcode ROM prior to when said each of said plurality of queue entries is provided to said register logic, whereby a first one of said second micro instructions is provided to said register logic when said first one of said second micro instructions is required by said register logic, and wherein said early access logic employs said microcode entry point when said microcode entry point is within a bottom queue entry, said bottom queue entry comprising one of said each of said plurality of micro instruction queue entries, and wherein said bottom queue entry will be provided to said register logic during a next clock cycle.

12. (Cancelled)

13. (Previously Presented) The apparatus as recited in claim 11, wherein said plurality of queue entries comprises four queue entries.

14. (Original) The apparatus as recited in claim 11, wherein said first micro instructions comprise three micro instructions.

15. (Original) The apparatus as recited in claim 14, wherein the microcode ROM access delay comprises four clock cycles.

16. (Cancelled)

17. (Previously Presented) The apparatus as recited in claim 11, wherein said translator is configured to generate said each of said plurality of micro instruction queue entries.

18. (Previously Presented) The apparatus as recited in claim 17, wherein said translator is configured to provide a generated queue entry to a top queue entry, wherein said top queue entry comprises one of said each of said plurality of micro instruction queue entries.

19. (Original) The apparatus as recited in claim 17, wherein said translator is configured to provide a generated queue entry to a mux, and, when said plurality of queue entries is empty, said mux provides said generated queue entry to said register logic during a next clock cycle.

20. (Original) The apparatus as recited in claim 19, wherein said early access logic employs a bypass microcode entry point corresponding to said generated micro instruction queue entry.

21. (Currently Amended) A method for precluding microprocessor pipeline stalls resulting from microcode ROM access delay, the method comprising:

translating instructions and first providing first parts of micro instruction sequences within a corresponding each of a plurality of micro instruction queue entries in order to a micro instruction queue, and second providing the corresponding each of a plurality of micro instruction queue entries to register logic in the order;

storing second parts of the micro instruction sequences within a microcode ROM;

obtaining a microcode entry point from within one of ~~said plurality~~the plurality of micro instruction queue entries in the micro instruction queue, the one of the plurality of micro instruction queue entries comprising ~~first micro instructions~~one of the first parts of the micro instruction sequences;

employing the microcode entry point to access one of the second parts of the micro instruction sequences~~second micro instructions within a within the~~ microcode ROM, wherein said employing is performed prior to when the one of the plurality of micro instruction queue entries is routed to the register logic in a following pipeline stage, and whereby said employing enables the second micro instructions to be provided to the following pipeline stage without incurring the microprocessor pipeline stalls; and

selecting the microcode entry point from within a bottom micro instruction queue entry, the bottom micro instruction queue entry comprising the one of a plurality of micro instruction queue entries, wherein the bottom micro instruction queue entry will be provided to the following pipeline stage during a next ~~clock cycle~~clock cycle.

22. (Original) The method as recited in claim 21, further comprising:

issuing the plurality of micro instruction queue entries in order to the following pipeline stage.

23. (Original) The method as recited in claim 22, wherein the plurality of micro instruction queue entries comprises four micro instruction queue entries.

24. (Original) The method as recited in claim 21, wherein the first micro instructions comprises three micro instructions.

25. (Original) The method as recited in claim 24, wherein the microcode ROM access delay comprises four clock cycles.

26. (Cancelled)

27. (Original) The method as recited in claim 21, further comprising:

generating a current microcode entry point as part of said translating.

28. (Original) The method as recited in claim 27, wherein said generating provides the current microcode entry point to a top micro instruction queue entry, and wherein the top micro instruction queue entry comprises another of the plurality of micro instruction queue entries.

29. (Original) The method as recited in claim 27, wherein said generating provides the current microcode entry point to a mux, and, when the plurality of micro instruction queue entries is empty, the mux provides the current microcode entry point to the following pipeline stage during a next clock cycle.

REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1, 3-5, 8-11, 13-15, 17-25, and 27-29 are pending in the application. The Examiner additionally stated that claims 1, 3-5, 8-11, 13-15, 17-25, and 27-29 are rejected. By this amendment, claims 1, 11, and 21 are amended. Hence, claims 1, 3-5, 8-11, 13-15, 17-25, and 27-29 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Claims

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claims 1, 11, and 21 under 35 U.S.C. 103(a) as being unpatentable over the background of the specification in view of Philip, U.S. Patent No. 3,130,387 (hereinafter, "Philip"). Applicant respectfully traverses the Examiner's rejections.

The Examiner noted that the background of the instant application specification teaches a microprocessor apparatus, for precluding a pipeline stall due to microcode ROM access delay, the microprocessor apparatus comprising:

- a translator, configured to generate a plurality of micro instructions corresponding to an instruction and a microcode entry point (Spec: 0007); and
- early access logic, couple to said translator, configured to employ said microcode entry point to access a microcode ROM prior to when said microcode entry point is provided to register logic, whereby said microcode ROM provides a first micro instruction to said register logic when said first micro instruction is required by said register logic, and wherein said translator provides said plurality of micro instructions to said register logic (Spec: 0009) (The Examiner stated that all of this inherently flows from the specification and that it would be inefficient to have another unit generate the ROM address when the translator is already determining what the instruction is.)

The Examiner stipulated that the only aspect that is not taught by the specification is a buffer between the translator and the register logic and that the inventors have identified the problem that exists in the dual translation process described in 0009 of the specification is that the translator and the ROM can get out of synchronization because of delay caused by a larger ROM. The Examiner opined that the problem described is a very old one that has been resolved long ago (Philip: Title), and that one of ordinary skill in the pertinent art would have readily realized that when two independent data producing objects become unsynchronized, that a buffer may be implemented.

The Examiner rejected claims 11 and 21 for the same reasons.

In reply, Applicant respectfully directs the Examiner's attention to the background of the specification, paragraph [0009], which is repeated below for ease of reference.

> **[0009]** For this reason, it is not uncommon in the art to find that a combination of the above two techniques is employed to translate instructions into corresponding micro instruction sequences. For instance, direct translation is often provided to translate those instructions having micro instruction sequences consisting of number of micro instructions that can be directly translated within the delay that would be otherwise experienced by providing those instructions to a microcode ROM. And for those instructions that have a corresponding number of micro instructions that are greater than what could be generated during the microcode ROM access delay time, an address is provided to the microcode ROM for a second part of a corresponding micro instruction sequence while at the same time a first part of the micro instruction sequence is directly translated. Consequently, during the clock cycle following when the last instruction of the first part of the micro instruction sequence is directly generated, a first instruction from the second part of the micro instruction sequence is provided by the microcode ROM. In this manner, the access delay of the microcode ROM is effectively absorbed by direct translation and instruction translation efficiency of an associated microprocessor is maximized.